A Branch-and-Bound Algorithm for Quadratically-Constrained Sparse Filter Design

Dennis Wei and Alan V. Oppenheim

***Abstract*—This paper presents an exact algorithm for sparse filter design under a quadratic constraint on filter performance. The algorithm is based on branch-and-bound, a combinatorial optimization procedure that can either guarantee an optimal solution or produce a sparse solution with a bound on its deviation from optimality. To reduce the complexity of branch-and-bound, several methods are developed for bounding the optimal filter cost. Bounds based on infeasibility yield incrementally accumulating improvements with minimal computation, while two convex relaxations, referred to as linear and diagonal relaxations, are derived to provide stronger bounds. The approximation properties of the two relaxations are characterized analytically as well as numerically. Design examples involving wireless channel equalization and minimum-variance distortionless-response beamforming show that the complexity of obtaining certifiably optimal solutions can often be significantly reduced by incorporating diagonal relaxations, especially in more difficult instances. In the case of early termination due to computational constraints, diagonal relaxations strengthen the bound on the proximity of the final solution to the optimum.**

I. Introduction

The cost of a discrete-time filter implementation is often largely determined by the number of arithmetic operations. Accordingly, sparse filters, i.e., filters with relatively few non-zero coefficients, offer a means to reduce cost, especially in hardware implementations. Sparse filter design has been investigated by numerous researchers in the context of frequency response approximation [1]–[4], communication channel equalization [5]–[10], speech coding [11], and signal detection [12].

In a companion paper [13], we formulate a problem of designing filters of maximal sparsity subject to a quadratic constraint on filter performance. We show that this general formulation encompasses the problems of least-squares frequency-response approximation, mean square error estimation, and signal detection. The focus in [13] is on low-complexity algorithms for solving the resulting combinatorial optimization problem. Such algorithms are desirable when computation is limited, for example in adaptive design. When the quadratic constraint has special structure, low-complexity algorithms are sufficient to guarantee optimally sparse designs. For the general case, a backward greedy selection algorithm is shown empirically to yield optimal or near-optimal solutions in many instances. We refer the reader to [13] for additional background on sparse filter design and a more detailed bibliography.

A major shortcoming of many low-complexity methods, including the backward selection algorithm in [13] and others (e.g. [3], [4], [8]–[10]), is that they do not indicate how close the resulting designs are to the true optimum. In the present paper, we take a different approach to address this shortcoming, specifically by combining branch-and-bound [14], an exact procedure for combinatorial optimization, with several methods for obtaining lower bounds on the optimal cost, i.e., bounds on the smallest feasible number of non-zero coefficients. The resulting algorithm maintains both a solution to the problem as well as a bound on its deviation from optimality. The algorithm in the current paper can therefore be seen as complementary to low-complexity algorithms that do not come with such guarantees.

One motivation for exact algorithms is to provide certifiably optimal solutions. In applications such as array design where the fabrication and operation of array elements can be very expensive, the guarantee of maximally sparse designs is especially attractive. Perhaps more importantly, exact algorithms are valuable as benchmarks for assessing the performance of lower-complexity algorithms that are often used in practice. One example of this is the use of the Wiener filter as the benchmark in adaptive filtering [15]. In the present context, we have used the algorithm in this paper to evaluate the backward selection algorithm in [13], showing that the latter often produces optimal or near-optimal solutions.

Given the complexity of combinatorial optimization problems such as sparse filter design, there are inevitably problem instances that are too large or difficult to be solved to optimality within the computational constraints of the application. In this setting, branch-and-bound can offer an appealing alternative. The algorithm can be terminated early, for example after a specified period of time, yielding both a feasible solution as well as a bound on its proximity to the optimum.

The challenge with branch-and-bound, whether run to completion or terminated early, is the combinatorial complexity of the problem. In this paper, we address the complexity by focusing on developing lower bounds on the optimal cost. While branch-and-bound algorithms have been proposed for sparse filter design [1], [2], [5], the determination of bounds does not appear to have received much attention;

Manuscript received April 14, 2012; accepted October 09, 2012. This work was supported in part by the Texas Instruments Leadership University Program.

D. Wei is with the Department of Electrical Engineering and Computer Science, University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109 USA; e-mail: dlwei@eecs.umich.edu.

A. V. Oppenheim is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Room 36-615, 77 Massachusetts Avenue, Cambridge, MA, 02139 USA; e-mail: avo@mit.edu.

the bounds used in [2], [5] are elementary, while [1] relies on the general-purpose solver CPLEX [16] which does not exploit the specifics of the sparse filter design problem. As we discuss in Section II, strong and efficiently computable bounds can be instrumental in mitigating the combinatorial nature of branch-and-bound. Design experiments show that the bounding techniques in this paper can dramatically decrease complexity, by orders of magnitude in difficult instances, and even when our MATLAB implementation is compared to sophisticated commercial software such as CPLEX. In the case of early termination, the proposed techniques lead to stronger guarantees on the final solution.

Three classes of bounds are discussed. Bounds based on infeasibility require minimal computation and can be easily applied to every branch-and-bound subproblem, but are consequently rather weak. To derive stronger bounds, we consider relaxations of the sparse design problem that can be solved efficiently. The first relaxation, referred to as linear relaxation [14], is a common technique in integer optimization adapted to our problem. The second relaxation exploits the simplicity of the problem when the matrix defining the quadratic constraint is diagonal, as discussed in [13]. For the non-diagonal case, we propose an optimized diagonal approximation referred to as a diagonal relaxation. The approximation properties of the two relaxations are analyzed to gain insight into when diagonal relaxations in particular are expected to give strong bounds. Numerical experiments complement the analysis and demonstrate that diagonal relaxations are tighter than linear relaxations under a range of conditions. Using the channel equalization and beamforming examples from [13], it is shown that diagonal relaxations can greatly reduce the time required to solve an instance to completion, or else give tighter bounds when the algorithm is terminated early.

The basic optimization problem addressed in this paper is the same as in [13], and hence we make reference throughout the current paper to results already derived in [13]. We emphasize however that the two papers take fundamentally different approaches: [13] focuses on low-complexity algorithms that ensure optimal designs in special cases but not in the general case, whereas the current paper presents an exact algorithm for the general case as well as methods for bounding the deviation from optimality. We also note that the linear and diagonal relaxations were introduced in a preliminary publication [17]. The current paper significantly extends [17] by including additional analytical and numerical results pertaining to the relaxations, presenting a branch-and-bound algorithm that incorporates the relaxations as well as lower-complexity bounds, and demonstrating improved computational complexity in solving sparse filter design problems.

The remainder of the paper proceeds as follows. In Section II, we state the problem of quadratically-constrained sparse filter design, review the branch-and-bound method for solving such combinatorial optimization problems, and introduce our proposed algorithm. In Section III, several methods for obtaining lower bounds are discussed, beginning with low-complexity bounds based on infeasibility and proceeding to linear and diagonal relaxations, together with an analysis of approximation properties and a numerical comparison. The branch-and-bound algorithm is applied to filter design examples in Section IV to illustrate the achievable complexity reductions.

II. Problem statement and branch-and-bound solution

As in [13], we consider the problem of minimizing the number of non-zero coefficients in an FIR filter of length N subject to a quadratic constraint on filter performance, i.e.,

$$\min_{\mathbf{b}} \quad \|\mathbf{b}\|_0 \qquad \text{s.t.} \qquad (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma, \quad (1)$$

where the zero-norm $\|\mathbf{b}\|_0$ denotes the number of non-zero components in the coefficient vector $\mathbf{b}$, $\mathbf{c}$ is a vector representing the solution that maximizes performance without regard to sparsity, $\mathbf{Q}$ is a symmetric positive definite matrix corresponding to the performance criterion, and γ is a positive constant. As discussed in [13], several variations of the sparse filter design problem can be reduced to (1). The quadratic constraint in (1) may be interpreted geometrically as specifying an ellipsoid, denoted as $\mathcal{E}_{\mathbf{Q}}$, centered at $\mathbf{c}$. As illustrated in Fig. 1, the eigenvectors and eigenvalues of $\mathbf{Q}$ determine the orientation and relative lengths of the axes of $\mathcal{E}_{\mathbf{Q}}$ while γ determines its absolute size. We will make reference to this ellipsoidal interpretation in Section III.

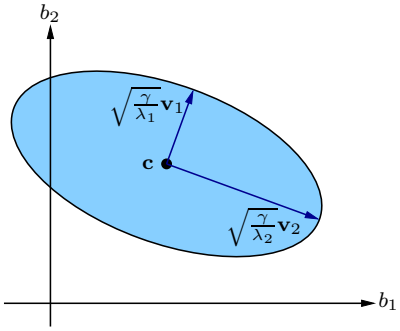

Fig. 1. Ellipsoid $\mathcal{E}_{\mathbf{Q}}$ formed by feasible solutions to problem (1). λ_1 and λ_2 are eigenvalues of $\mathbf{Q}$ and $\mathbf{v}_1$ and $\mathbf{v}_2$ are the associated eigenvectors.

Solving problem (1) generally requires combinatorial optimization, although certain special cases permit much more efficient algorithms as seen in [13]. In this section, we review the branch-and-bound procedure for combinatorial optimization with emphasis on the role of bounds in reducing complexity. Further background on branch-and-bound can be found in [14]. We then present our specific branch-and-bound algorithm for solving (1).

For convenience and for later use in Section III-B, problem (1) is reformulated as a mixed integer optimization problem. To each coefficient b_n we associate a binary-valued indicator variable i_n with the property that $i_n = 0$ if $b_n = 0$ and $i_n = 1$ otherwise. The sum of the indicator variables is therefore equal

to $\|\mathbf{b}\|_0$ and (1) can be restated as follows:

$$\begin{aligned} \min_{\mathbf{b},\mathbf{i}} \quad & \sum_{n=1}^{N} i_n \\ \text{s.t.} \quad & (\mathbf{b}-\mathbf{c})^T \mathbf{Q} (\mathbf{b}-\mathbf{c}) \le \gamma, \\ & |b_n| \le B_n i_n \quad \forall\, n, \\ & i_n \in \{0,1\} \quad \forall\, n, \end{aligned} \tag{2}$$

where B_n, $n = 1, \ldots, N$, are positive constants. The second constraint in (2) ensures that i_n serves as an indicator, forcing b_n to zero when $i_n = 0$. When $i_n = 1$, the second constraint becomes a bound on the absolute value of b_n. The constants B_n are chosen large enough so that these bounds on $|b_n|$ do not further restrict the set of feasible $\mathbf{b}$ from that in (1). Specific values for B_n will be chosen later in Section III-B in the context of linear relaxation.

The branch-and-bound procedure solves problem (2) by recursively dividing it into subproblems with fewer variables. The first two subproblems are formed by selecting an indicator variable and fixing it to zero in the first subproblem and to one in the second. Each of the two subproblems, if not solved directly, is subdivided into two more subproblems by fixing a second indicator variable. This process, referred to as branching, produces a binary tree of subproblems as depicted in Fig. 2.

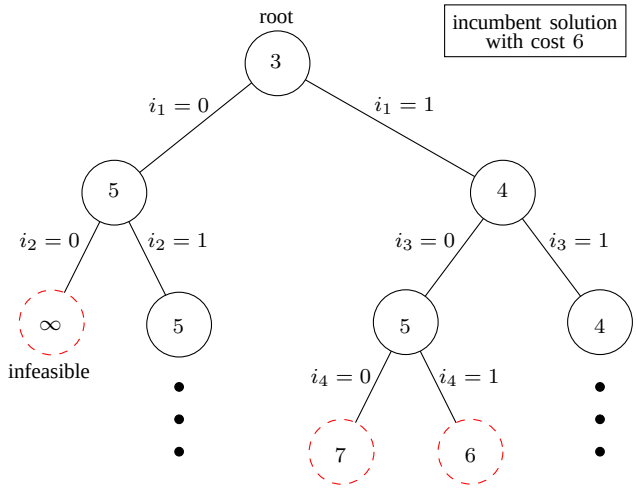

Fig. 2. Example of a branch-and-bound tree. Each circle represents a subproblem and the branch labels indicate the indicator variables that are fixed in going from a parent to a child. The number in each circle is a lower bound on the optimal cost of the corresponding subproblem. Given an incumbent solution with a cost of 6, the subproblems marked by dashed circles need not be considered any further.

Each subproblem is defined by three index sets, a set $\mathcal{Z} = \{n : i_n = 0\}$ corresponding to coefficients constrained to a value of zero, a set $\mathcal{U} = \{n : i_n = 1\}$ of coefficients assumed to be non-zero, and a set $\mathcal{F}$ consisting of the remainder. As shown in [13], a subproblem thus defined is equivalent to the following problem:

$$\begin{aligned} \min_{\mathbf{b}_\mathcal{F}} \quad & |\mathcal{U}| + \|\mathbf{b}_\mathcal{F}\|_0 \\ \text{s.t.} \quad & \left(\mathbf{b}_\mathcal{F} - \mathbf{c}_{\text{eff}}\right)^T \mathbf{Q}_{\text{eff}} \left(\mathbf{b}_\mathcal{F} - \mathbf{c}_{\text{eff}}\right) \le \gamma_{\text{eff}}, \end{aligned} \tag{3}$$

where $\mathbf{b}_\mathcal{F}$ denotes the $|\mathcal{F}|$-dimensional subvector of $\mathbf{b}$ indexed by $\mathcal{F}$ (similarly for other vectors). Problem (3) is an $|\mathcal{F}|$-dimensional instance of the original problem (1) with effective parameters given by

$$\mathbf{Q}_{\text{eff}} = \mathbf{Q}_{\mathcal{F}\mathcal{F}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}} \left(\mathbf{Q}_{\mathcal{U}\mathcal{U}}\right)^{-1} \mathbf{Q}_{\mathcal{U}\mathcal{F}}, \tag{4a}$$

$$\mathbf{c}_{\text{eff}} = \mathbf{c}_\mathcal{F} + \left(\mathbf{Q}_{\text{eff}}\right)^{-1} \left(\mathbf{Q}_{\mathcal{F}\mathcal{Z}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}} (\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1} \mathbf{Q}_{\mathcal{U}\mathcal{Z}}\right) \mathbf{c}_\mathcal{Z}, \tag{4b}$$

$$\gamma_{\text{eff}} = \gamma - \mathbf{c}_\mathcal{Z}^T \left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{Z}\mathcal{Z}}\right)^{-1} \mathbf{c}_\mathcal{Z}, \tag{4c}$$

where $\mathbf{Q}_{\mathcal{F}\mathcal{U}}$ denotes the submatrix of $\mathbf{Q}$ with rows indexed by $\mathcal{F}$ and columns indexed by $\mathcal{U}$ (similarly for other matrices). This reduced-dimensionality formulation leads to greater efficiency in the branch-and-bound algorithm. Furthermore, the common structure allows every subproblem to be treated in the same way.

The creation of subproblems through branching is complemented by the computation of lower bounds on the optimal cost in (3) for subproblems that are not solved directly. Infeasible subproblems can be regarded as having a lower bound of $+\infty$. Since a child subproblem is related to its parent by the addition of one constraint, the lower bound for the child must be at least as large as that for the parent. This non-decreasing property of the lower bounds is illustrated in Fig. 2. In addition, feasible solutions may also be obtained for certain subproblems. The algorithm keeps a record of the feasible solution with the lowest cost thus far, referred to as the incumbent solution. It is apparent that if the lower bound for a subproblem is equal to or higher than the cost of the incumbent solution, then the subproblem cannot lead to better solutions and can thus be eliminated from the tree along with all of its descendants. This pruning operation is also illustrated in Fig. 2. To minimize complexity, it is clearly desirable to prune as many subproblems as possible.

Although in worst-case examples the complexity of branch-and-bound remains exponential in N [14], for more typical instances the situation can be greatly improved. One important contributor to greater efficiency is an initial incumbent solution that is already optimal or nearly so. Such a solution allows for more subproblems to be pruned compared to an incumbent solution with higher cost. Good initial solutions can often be provided by heuristic algorithms.

The determination of lower bounds on the other hand is a more difficult and less studied problem. The availability and quality of subproblem lower bounds also has a strong impact on the complexity of branch-and-bound. As with near-optimal incumbent solutions, stronger (i.e. larger) lower bounds result in more subproblems being pruned. Moreover, these lower bounds must be efficiently computable since they may be evaluated for a large number of subproblems. Section III discusses several bounding methods with computational efficiency in mind.

We now introduce our proposed algorithm for solving (1). A summary is provided in Fig. 3 with certain steps numbered for convenient reference. The algorithm is initialized by generating an incumbent solution $\mathbf{b}_I$ using the backward greedy algorithm of [13]. Other initializations could also be used with no effect on the final solution if the algorithm is

Input: Parameters $\mathbf{Q}$, $\mathbf{c}$, γ
Output: Optimal solution $\mathbf{b}_I$ to (1)
Initialize: Generate incumbent solution $\mathbf{b}_I$ using backward greedy algorithm of [13]. Place root problem in list with $LB = 0$.
while list not empty **do**
1) Select subproblem with minimal LB and remove from list. Subproblem parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, γ_{eff} given by (4).
if $i_{\text{last}} = 0$ **then**
2) Identify coefficients in $\mathcal{F}$ for which a zero value is no longer feasible using (6) (Section III-A). Update $\mathcal{U}$, $\mathcal{F}$, $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$ if necessary.
if $|\mathcal{U}| \geq \|\mathbf{b}_I\|_0$ **then**
Prune current subproblem, go to step 1.
if $LB < |\mathcal{U}| + 2$ **then**
3) Check for solutions with $\|\mathbf{b}_{\mathcal{F}}\|_0 = 0$, $\|\mathbf{b}_{\mathcal{F}}\|_0 = 1$ (Section III-A).
if subproblem solved **and** $|\mathcal{U}| + \|\mathbf{b}_{\mathcal{F}}\|_0 < \|\mathbf{b}_I\|_0$ **then**
Update $\mathbf{b}_I$ and prune list. Go to step 1.
else
$LB \leftarrow |\mathcal{U}| + 2$.
if $LB \geq \|\mathbf{b}_I\|_0$ **then**
Prune current subproblem, go to step 1.
4) Generate feasible solution $\mathbf{b}_{\mathcal{F}}$ with $\|\mathbf{b}_{\mathcal{F}}\|_0 = |\mathcal{F}| - 1$.
if $|\mathcal{U}| + |\mathcal{F}| - 1 < \|\mathbf{b}_I\|_0$ **then**
Update $\mathbf{b}_I$ and prune list (possibly including current subproblem).
if $i_{\text{last}} = 0$ **and** $|\mathcal{F}| \geq N_{\text{relax}} \approx 20$ **then**
5) Solve linear or diagonal relaxation (Sections III-B, III-C) and update LB.
if $LB \geq \|\mathbf{b}_I\|_0$ **then**
Prune current subproblem, go to step 1.
6) Create two new subproblems by fixing i_m to $0, 1$, where m is given by (5). Go to step 1.

Fig. 3. Branch-and-bound algorithm

run to completion; however, the amount of pruning and hence the rate of convergence would decrease with an inferior initial solution. The algorithm uses a list to track subproblems in the branch-and-bound tree that are open in the sense of having lower bounds (denoted as LB in Fig. 3) that are less than the incumbent cost. In each iteration, an open subproblem is selected and processed in an attempt to improve the lower bound inherited from its parent. Pruning results as soon as the lower bound rises above the incumbent cost, a condition that is checked at several points. Feasible solutions are also generated and may occasionally trigger updates to the incumbent solution and pruning based on the new incumbent cost. A subproblem that is not solved or pruned leads to branching and the addition of two subproblems to the list. The algorithm terminates when the list is empty; alternatively, it can be terminated early after a specified period of time or number of subproblems processed.

In Step 1, we choose an open subproblem for which the current lower bound is among the lowest. This choice yields the fastest possible increase in the global lower bound, i.e., the minimum of the lower bounds among open subproblems. Thus if the algorithm is terminated early, the bound on the deviation from optimality of the incumbent solution is as tight as possible. Furthermore, it is prudent to defer on subproblems with the highest lower bounds since these are the first to be pruned whenever the incumbent solution is improved.

Steps 2–5 relate to the updating of lower bounds and are discussed further in Section III. The indicator variable i_{last} refers to the last indicator variable that was fixed in creating a subproblem from its parent. We note for now that solving relaxations is by far the most computationally intensive step and is therefore justified only if a sufficient number of subproblems can be pruned as a result. We have found that it is not worthwhile to solve relaxations of subproblems for which $i_{\text{last}} = 1$ since they rarely lead to pruning. In addition, small subproblems can often be solved more efficiently by relying only on the low-complexity steps 2 and 3 and the branch-and-bound process. For this reason, we solve relaxations only when the subproblem dimension $|\mathcal{F}|$ equals or exceeds a parameter N_{relax}. The best value of N_{relax} depends on the complexity of solving relaxations relative to running branch-and-bound without relaxations. In our experiments, we have found $N_{\text{relax}} \approx 20$ to be a good choice.

In Step 6, we choose the index m for branching according to

$$m = \arg\min_{n \in \mathcal{F}} \quad \gamma - \frac{c_n^2}{\left(\mathbf{Q}^{-1}\right)_{nn}}, \tag{5}$$

which results in the smallest possible (but still positive) value for the parameter γ_{eff} in the $i_m = 0$ child subproblem. Thus the $i_m = 0$ subproblem, while still feasible, tends to be severely constrained and the subtree created under the parent is unbalanced with many more nodes under the $i_m = 1$ branch than under the $i_m = 0$ branch. Generally speaking, the higher that these asymmetric branchings occur in the tree, the greater the reduction in the number of subproblems. In the extreme case, if one of the branches under the root problem supports very few feasible subproblems, the number of subproblems is almost halved. We have observed that this branching rule tends to reduce the number of subproblems in agreement with the above intuition.

III. Approaches to bounding the optimal cost

In this section, we discuss the determination of lower bounds on the optimal cost of problem (1), beginning in Section III-A with bounds that are inexpensive to compute and continuing in Sections III-B and III-C with two convex relaxations of problem (1) that lead to stronger lower bounds. The two relaxations are evaluated and compared numerically in Section III-D. While our presentation will focus on the root problem (1), all of the techniques are equally applicable to any subproblem by virtue of the common structure noted in Section II.

A. Bounds based on infeasibility

We begin with two methods based on infeasibility, corresponding to Steps 2 and 3 in Fig. 3. While the resulting bounds tend to be weak when used in isolation, they become more powerful as part of a branch-and-bound algorithm where they

can be applied inexpensively to each new subproblem, improving lower bounds incrementally as the algorithm descends the tree.

For a subproblem specified by index sets $(\mathcal{Z},\mathcal{U},\mathcal{F})$ as defined in Section II, the number of elements in $\mathcal{U}$ is clearly a lower bound on the optimal cost in (3). This lower bound may be improved and the subproblem dimension reduced by identifying those coefficients in $\mathcal{F}$ for which a value of zero is no longer feasible (Step 2 in Fig. 3). As derived in [13], setting $b_n = 0$ is feasible for the root problem (1) if and only if

$$\frac{c_n^2}{\left(\mathbf{Q}^{-1}\right)_{nn}} \leq \gamma. \tag{6}$$

A similar condition stated in terms of the effective parameters in (4) holds for an arbitrary subproblem. We set $i_n = 1$ for indices $n \in \mathcal{F}$ for which (6) is not satisfied, thus increasing $|\mathcal{U}|$ and decreasing $|\mathcal{F}|$. In terms of the branch-and-bound tree, this corresponds to eliminating infeasible $i_n = 0$ branches. The increase in $|\mathcal{U}|$ and corresponding reduction in dimension can be significant if γ is relatively small so that (6) is violated for many indices n.

For the remainder of the paper we will assume that the above test is performed on every subproblem and variables are eliminated as appropriate. Thus we need only consider subproblems for which (6) is satisfied for all $n \in \mathcal{F}$, i.e., a feasible solution exists whenever a single coefficient is constrained to zero. This fact is used in Step 4 in Fig. 3 to generate feasible solutions to subproblems with $\|\mathbf{b}_{\mathcal{F}}\|_0 = |\mathcal{F}| - 1$, where the single zero-valued coefficient is chosen to maximize the margin in (6). Furthermore, as indicated in Fig. 3, it is not necessary to perform the test on subproblems for which $i_{\text{last}} = 1$. Setting $i_{\text{last}} = 1$ does not change the set of feasible $\mathbf{b}$, and consequently any coefficient for which a value of zero is feasible in the parent subproblem retains that property in the child subproblem.

It is possible to generalize the test to identify larger subsets of coefficients that cannot yield feasible solutions when simultaneously constrained to zero. However, the required computation increases dramatically because the number of subsets grows rapidly with subset size and because the generalization of condition (6) requires matrix inversions of increasing complexity. Moreover, incorporating information from tests involving larger subsets is less straightforward than simply setting certain i_n to 1.

A second class of low-complexity lower bounds relies on determining whether solutions with small numbers of non-zero elements are infeasible (Step 3 in Fig. 3). In the extreme case, the solution $\mathbf{b} = \mathbf{0}$ is feasible if $\beta \equiv \gamma - \mathbf{c}^T\mathbf{Q}\mathbf{c} \geq 0$. Hence a negative β implies a lower bound of at least 1 ($|\mathcal{U}| + 1$ for a general subproblem) on the optimal cost. For the case of solutions with a single non-zero coefficient, the feasibility condition is

$$-\frac{f_n^2}{Q_{nn}} \leq \beta, \tag{7}$$

where the vector $\mathbf{f} = \mathbf{Q}\mathbf{c}$. Condition (7) is a special case of a general condition (equation (13) in [13]) for feasibility when only a subset of coefficients is permitted to be non-zero. If (7) is satisfied for some $n \in \mathcal{F}$, there exists a solution with b_n non-zero and the remaining coefficients equal to zero, and therefore the optimal cost is 1 provided that the solution $\mathbf{b} = \mathbf{0}$ has been excluded. Otherwise, we conclude that the optimal cost is no less than 2 ($|\mathcal{U}|+2$ in general). Since this test yields a lower bound of at most $|\mathcal{U}| + 2$, the execution of Step 3 in Fig. 3 depends on whether or not the inherited lower bound already exceeds $|\mathcal{U}| + 2$. The enumeration of solutions can be extended to larger subsets of coefficients, resulting in either an optimal solution or progressively higher lower bounds. The increase in computational effort however is the same as for generalizations of (6).

B. Linear relaxation

The lower bounds discussed in Section III-A are simple to compute but are only effective for pruning low-dimensional or severely constrained subproblems. Better bounds can be obtained through relaxations[1] of problem (1), constructed in such a way that their solutions yield lower bounds on the optimal cost of (1). As the term suggests, these relaxations are also intended to be significantly easier to solve than the original problem. In this subsection, we apply a common technique known as linear relaxation to (1) and consider its approximation properties. An alternative relaxation, referred to as diagonal relaxation, is developed in Section III-C.

To obtain a linear relaxation of problem (1), we start with its alternative formulation as a mixed integer optimization problem (2) and relax the binary constraints on i_n, allowing i_n to vary continuously between 0 and 1. The minimization may then be carried out in two stages. In the first stage, $\mathbf{b}$ is held constant while the objective is minimized with respect to $\mathbf{i}$, resulting in $i_n = |b_n| / B_n$ for each n. Substituting back into (2) gives the following minimization with respect to $\mathbf{b}$, which we refer to as a linear relaxation:

$$\min_{\mathbf{b}} \quad \sum_{n=1}^{N} \frac{|b_n|}{B_n} \qquad \text{s.t.} \qquad (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma. \tag{8}$$

Problem (8) is a quadratically-constrained weighted 1-norm minimization, a convex optimization problem that can be solved efficiently. Since the set of feasible indicator vectors $\mathbf{i}$ is enlarged in deriving (8) from (2), the optimal value of (8) is a lower bound on that of (2). More precisely, since the optimal value of (2) must be an integer, the ceiling of the optimal value of (8) is also a lower bound.

To maximize the optimal value of (8), thereby maximizing the lower bound on the optimal value of (2), the constants B_n in the objective function of (8) should be made as small as possible. Recall from Section II that B_n must also be large enough to leave the set of feasible $\mathbf{b}$ in (2) unchanged from that in (1), i.e., we require $B_n \geq |b_n|$ for all n whenever $\mathbf{b}$ satisfies the quadratic constraint in (1). These conditions imply that B_n should be chosen as

$$\begin{aligned} B_n^* &= \max\left\{|b_n| : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\} \\ &= \max\left\{B_n^{+*}, B_n^{-*}\right\}, \end{aligned} \tag{9}$$

[1]Following common usage in the field of optimization, we use the term relaxation to refer to both the technique used to relax certain constraints in a problem as well as the modified problem that results.

where

$$\begin{aligned} B_n^{\pm *} &= \max\left\{\pm b_n : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \le \gamma\right\} \\ &= \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} \pm c_n. \end{aligned} \tag{10}$$

The closed-form expressions for $B_n^{\pm *}$ are derived in [18, App. B.1]. Hence (9) simplifies to

$$B_n^* = \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} + |c_n|\,.$$

A still stronger lower bound on (2) can be obtained by first separating each coefficient b_n into its positive and negative parts b_n^+ and b_n^- as follows:

$$b_n = b_n^+ - b_n^-, \quad b_n^+,\, b_n^- \ge 0. \tag{11}$$

Under the condition that at least one of b_n^+, b_n^- is always zero, the representation in (11) is unique, $b_n = b_n^+$ for $b_n > 0$, and $b_n = -b_n^-$ for $b_n < 0$. By assigning to each pair b_n^+, b_n^- corresponding indicator variables i_n^+, i_n^- and positive constants B_n^+, B_n^-, a mixed integer optimization problem equivalent to (2) may be formulated (see [18, Sec. 3.3.1] for details). Applying linear relaxation as above to this alternative mixed integer formulation results in

$$\begin{aligned} \min_{\mathbf{b}^+,\mathbf{b}^-} \quad & \sum_{n=1}^{N}\left(\frac{b_n^+}{B_n^+} + \frac{b_n^-}{B_n^-}\right) \\ \text{s.t.} \quad & (\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c})^T\mathbf{Q}(\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c}) \le \gamma, \\ & \mathbf{b}^+ \ge \mathbf{0}, \quad \mathbf{b}^- \ge \mathbf{0}. \end{aligned} \tag{12}$$

Problem (12) is a quadratically constrained linear program and is also efficiently solvable. The smallest values for B_n^+ and B_n^- that ensure that (12) is a valid relaxation are given by B_n^{+*} and B_n^{-*} in (10). Using a standard linear programming technique based on the representation in (11) to replace the absolute value functions in (8) with linear functions (see [19]), it can be seen that (8) is a special case of (12) with $B_n^+ = B_n^- = B_n$. Since $B_n^* = \max\{B_n^{+*}, B_n^{-*}\}$, the optimal value of (12) with $B_n^\pm = B_n^{\pm *}$ is at least as large as that of (8) with $B_n = B_n^*$, and therefore (12) is at least as strong a relaxation as (8). Henceforth we will use the term linear relaxation to refer to (12) with $B_n^\pm = B_n^{\pm *}$.

In general, given a relaxation of an optimization problem, it is of interest to analyze the conditions under which the relaxation is either a good or a poor approximation to the original problem. The quality of approximation is often characterized by the approximation ratio, defined as the ratio of the optimal value of the relaxation to the optimal value of the original problem. In the case of the linear relaxation in (12), the quality of approximation can be understood geometrically. We first note that the cost function in (12) can be regarded as an asymmetrically-weighted 1-norm with different weights for positive and negative coefficient values. Recalling the ellipsoidal interpretation of the feasible set discussed in Section II, the minimization problem in (12) can be represented graphically as in Fig. 4. Note that our assumption that (6) is satisfied for all n implies that the ellipsoid $\mathcal{E}_\mathbf{Q}$ must intersect all of the coordinate planes; otherwise the problem dimension could be reduced. The asymmetric diamond shape represents a level contour of the 1-norm weighted by $1/B_n^{\pm *}$. As seen from (10), the weights $B_n^{\pm *}$ correspond to the maximum extent of $\mathcal{E}_\mathbf{Q}$ along the positive and negative coordinate directions and can be found graphically as indicated in Fig. 4. The solution to the weighted 1-norm minimization can be visualized by inflating the diamond until it just touches the ellipsoid. The optimal solution is given by the point of tangency and the optimal value by the tangent contour.

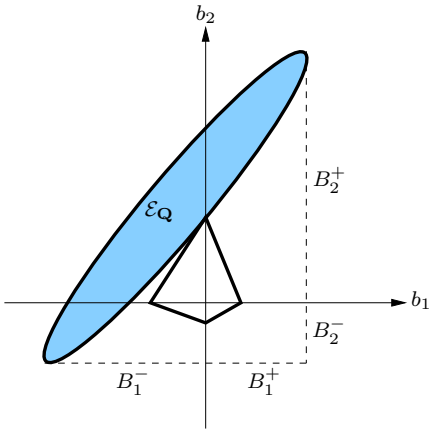

Fig. 4. Interpretation of the linear relaxation as a weighted 1-norm minimization and a graphical representation of its solution.

Based on the geometric intuition in Fig. 4, the optimal value of the linear relaxation and the resulting lower bound on (1) are maximized when the ellipsoid $\mathcal{E}_\mathbf{Q}$ is such that the ℓ_1 diamond can grow relatively unimpeded. This is the case for example if the major axis of $\mathcal{E}_\mathbf{Q}$ is oriented parallel to a level surface of the 1-norm and the remaining ellipsoid axes are very short. The algebraic equivalent in terms of the matrix $\mathbf{Q}$ is to have one eigenvalue that is much smaller than the others. The corresponding eigenvector should have components that are roughly half positive and half negative with magnitudes that conform to the weights $B_n^{\pm *}$. In [18, Sec. 3.3.2, App. B.3], it is shown that for instances constructed as just described, the optimal value of the linear relaxation is large enough to match the optimal cost of (1), i.e., the approximation ratio is equal to 1, the highest possible value. Hence there exist instances of (1) for which the linear relaxation is a tight approximation.

Conversely, the optimal value of the linear relaxation is small when the ellipsoid obstructs the growth of the ℓ_1 ball. This occurs if the major axis of $\mathcal{E}_\mathbf{Q}$ is oriented so that it points toward the origin, or equivalently in terms of $\mathbf{Q}$ if the eigenvector associated with the smallest eigenvalue is a multiple of the vector $\mathbf{c}$. It is shown in [18, Sec. 3.3.2, App. B.4] that instances with this property exhibit approximation ratios that are close to zero. The approximation ratio cannot be exactly equal to zero since that would require the optimal value of the linear relaxation to be zero, which occurs only if $\mathbf{b} = \mathbf{0}$ is a feasible solution to (1), i.e., only if the original optimal cost is also equal to zero. Therefore the worst case is for the linear relaxation to have an optimal value less than 1 (so that its ceiling is equal to 1) while the original problem has an optimal value equal to $N - 1$ (given our assumption that (6) is satisfied for all n, the original optimal cost is at most $N - 1$). As shown in [18], there exist instances in which both

conditions are achieved, yielding a poor approximation ratio of $1/(N-1)$.

The above discussion implies that the approximation ratio for the linear relaxation can range anywhere between 0 and 1, and thus it is not possible to place a non-trivial guarantee on the ratio that holds for all instances of (1). It is possible however to obtain an absolute upper bound on the optimal value of the linear relaxation in terms of N, the total number of coefficients. We use the fact that any feasible solution to the linear relaxation (12) provides an upper bound on its optimal value. Choosing $\mathbf{b}^+ - \mathbf{b}^- = \mathbf{c}$, i.e., $b_n^+ = c_n$, $b_n^- = 0$ for $c_n \geq 0$ and $b_n^+ = 0$, $b_n^- = |c_n|$ for $c_n < 0$ results in an upper bound of

$$\sum_{n:c_n>0} \frac{c_n}{B_n^{+*}} + \sum_{n:c_n<0} \frac{|c_n|}{B_n^{-*}} = \sum_{n=1}^{N} \frac{|c_n|}{\sqrt{\gamma\big(\mathbf{Q}^{-1}\big)_{nn}} + |c_n|}, \quad (13)$$

where we have used (10). Given the assumption that (6) is satisfied for all n, each of the fractions on the right-hand side of (13) is no greater than $1/2$, and consequently the optimal value of the linear relaxation can be no larger than $N/2$. This upper bound can be further reduced by the factor

$$\theta = 1 - \sqrt{\frac{\gamma}{\mathbf{c}^T\mathbf{Q}\mathbf{c}}}, \quad (14)$$

which corresponds to scaling the solution $\mathbf{b}^+ - \mathbf{b}^- = \mathbf{c}$, which is in the center of the feasible set, so that it lies on the boundary nearest the origin.

It is apparent from (13) that the lower bound resulting from the linear relaxation cannot be tight if the optimal cost in (1) is greater than $\lceil \theta N/2 \rceil$. We infer that it is unlikely for the linear relaxation to be a good approximation to (1) in most instances, since if it were, this would imply that the optimal cost in (1) is not much greater than $\theta N/2$ in most cases, a fact that is considered unlikely. The situation is exacerbated if the factor θ in (14) is small. This motivates the consideration of an alternative relaxation as we describe in Section III-C.

We note in closing that Lemaréchal and Oustry [20] have shown that a common semidefinite relaxation technique is equivalent to linear relaxation when applied to sparsity maximization problems such as (1). As a consequence, the properties of the linear relaxation (12) noted in this section also apply to this type of semidefinite relaxation.

C. Diagonal relaxation

As an alternative to linear relaxations, in this subsection we discuss relaxations of (1) in which the matrix $\mathbf{Q}$ is replaced by a diagonal matrix, an approach we refer to as diagonal relaxation. As discussed in [13], the sparse design problem is straightforward to solve in the diagonal case, thus making it attractive as a relaxation when $\mathbf{Q}$ is non-diagonal.

To obtain a diagonal relaxation, the quadratic constraint in (1) is replaced with a similar constraint involving a positive definite diagonal matrix $\mathbf{D}$:

$$(\mathbf{b}-\mathbf{c})^T\mathbf{D}(\mathbf{b}-\mathbf{c}) = \sum_{n=1}^{N} D_{nn}(b_n - c_n)^2 \leq \gamma. \quad (15)$$

Geometrically, constraint (15) specifies an ellipsoid, denoted as $\mathcal{E}_{\mathbf{D}}$, with axes that are aligned with the coordinate axes. Since the relaxation is intended to provide a lower bound for the original problem, we require that the coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ enclose the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$ so that minimizing over $\mathcal{E}_{\mathbf{D}}$ yields a lower bound on the minimum over $\mathcal{E}_{\mathbf{Q}}$. For simplicity, the two ellipsoids are assumed to be concentric. Then it can be shown [18, Sec. 3.4.1] that the nesting of the ellipsoids is equivalent to $\mathbf{Q}-\mathbf{D}$ being positive semidefinite, which we write as $\mathbf{Q}-\mathbf{D} \succeq \mathbf{0}$ or $\mathbf{Q} \succeq \mathbf{D}$.

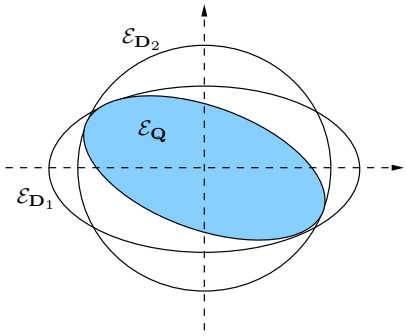

Fig. 5. Two different diagonal relaxations.

For every $\mathbf{D}$ satisfying $\mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q}$, minimizing $\|\mathbf{b}\|_0$ subject to (15) results in a lower bound for problem (1). Thus the set of diagonal relaxations is parameterized by $\mathbf{D}$ as shown in Fig. 5. As with linear relaxations in Section III-B, we are interested in finding a diagonal relaxation that is as tight as possible, i.e., a matrix $\mathbf{D}_d$ such that the minimum zero-norm associated with $\mathbf{D}_d$ is maximal among all valid choices of $\mathbf{D}$. To obtain such a relaxation, we make use of the following condition derived in [13], which specifies when constraint (15) admits a feasible solution $\mathbf{b}$ with K zero-valued elements:

$$S_K\big(\{D_{nn}c_n^2\}\big) \leq \gamma, \quad (16)$$

where $S_K\big(\{D_{nn}c_n^2\}\big)$ denotes the sum of the K smallest elements of the sequence $D_{nn}c_n^2$, $n = 1, \ldots, N$. Based on (16), the tightest diagonal relaxation may be determined by solving the following optimization:

$$\begin{aligned} E_d(K) = \max_{\mathbf{D}} \quad & S_K\big(\{D_{nn}c_n^2\}\big) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q}, \quad \mathbf{D} \text{ diagonal}, \end{aligned} \quad (17)$$

for values of K increasing from zero. If the optimal value $E_d(K)$ is less than or equal to γ, then condition (16) holds for every $\mathbf{D}$ satisfying the constraints in (17), and consequently a feasible solution $\mathbf{b}$ with K zero-valued coefficients exists for every such $\mathbf{D}$. We conclude that the minimum zero-norm in every diagonal relaxation can be at most $N-K$. The value of K is then incremented by 1 and (17) is re-solved. If on the other hand $E_d(K)$ is greater than γ for some $K = K_d + 1$, then according to (16) there exists a $\mathbf{D}_d$ for which it is not feasible to have a solution with $K_d + 1$ zero coefficients. When combined with the conclusions drawn for $K \leq K_d$, this implies that the minimum zero-norm with $\mathbf{D} = \mathbf{D}_d$ is equal to $N - K_d$. It follows that $N - K_d$ is the tightest lower bound achievable with a diagonal relaxation.

The foregoing procedure determines both the tightest possible diagonal relaxation and its optimal value at the same time. For convenience, we will refer to the overall procedure as solving the diagonal relaxation. The term diagonal relaxation will refer henceforth to the tightest diagonal relaxation.

The main computational burden in solving the diagonal relaxation lies in solving (17) for multiple values of K. It is shown in [18, Sec. 3.5.3] that (17) can be recast as the following semidefinite optimization problem in a scalar variable y_0 and vector variables $\mathbf{v}$ and $\mathbf{w}$:

$$\begin{aligned} \max_{y_0,\mathbf{v},\mathbf{w}} \quad & Ky_0 + \sum_{n=1}^{N} v_n \\ \text{s.t.} \quad & \mathbf{0} \preceq y_0\mathbf{I} + \mathrm{Diag}(\mathbf{w}) \preceq \mathrm{Diag}(\mathbf{c})\mathbf{Q}\,\mathrm{Diag}(\mathbf{c}), \\ & \mathbf{w} - \mathbf{v} \geq \mathbf{0}, \quad \mathbf{v} \leq \mathbf{0}, \end{aligned} \tag{18}$$

where $\mathrm{Diag}(\mathbf{x})$ denotes a diagonal matrix with the entries of $\mathbf{x}$ along the diagonal. The semidefinite reformulation (18) can be solved efficiently using interior-point algorithms. Further efficiency enhancements can be made as detailed in [18, Sec. 3.5]. For example, the monotonicity of the cost function in (17) with respect to K permits a binary search over K instead of the linear search discussed earlier.

As with the linear relaxation in Section III-B, it is of interest to understand how well the diagonal relaxation can approximate the original problem. It is clear that if $\mathbf{Q}$ is already diagonal, the diagonal relaxation and the original problem coincide and the approximation ratio defined in Section III-B is equal to 1. Based on Fig. 5, we would also expect the diagonal relaxation to yield a poor approximation when the original ellipsoid $\mathcal{E}_\mathbf{Q}$ is far from being coordinate-aligned. For example, $\mathcal{E}_\mathbf{Q}$ may be dominated by a single long axis with equal components in all coordinate directions, thus forcing the coordinate-aligned enclosing ellipsoid $\mathcal{E}_\mathbf{D}$ to be much larger than $\mathcal{E}_\mathbf{Q}$. This situation corresponds algebraically to $\mathbf{Q}$ having one eigenvalue that is much smaller than the rest, with the associated eigenvector having components of equal magnitude. In [18, Sec. 3.4.2], it is shown that when the smallest eigenvalue of $\mathbf{Q}$ is small enough, the diagonal relaxation has an optimal cost of zero while the original problem has a non-zero optimal cost. Thus the approximation ratio for the diagonal relaxation can range anywhere between 0 and 1, as with the linear relaxation. Furthermore, one class of instances for which the diagonal relaxation has a zero optimal cost is the same class for which the linear relaxation is a tight approximation. Hence there is no strict dominance relationship between the two relaxations (diagonal relaxations are clearly dominant in the case of diagonal $\mathbf{Q}$).

The above conclusions however are based on extreme instances, both best-case and worst-case. In more typical instances, the diagonal relaxation often yields a significantly better approximation than the linear relaxation. Several such cases are illustrated numerically in Section III-D. It has also been our experience as reported in Section IV that the diagonal relaxation provides strong bounds for problem instances encountered in applications of sparse filter design. We are thus motivated to understand from a theoretical perspective the situations in which the diagonal relaxation is expected to perform favorably. In the remainder of this subsection, we consider three restricted classes of instances and summarize our analytical results characterizing the approximation quality of the diagonal relaxation in these cases.

To state our results, we define K^* to be the maximum number of zero-valued coefficients in problem (1) (i.e., N minus the minimum zero-norm), and K_d to be the maximum number of zero-valued coefficients in the diagonal relaxation of (1). The enclosing condition $\mathcal{E}_\mathbf{Q} \subseteq \mathcal{E}_\mathbf{D}$ ensures that K_d is an upper bound on K^*. The ratio K_d/K^* is thus an alternative definition of approximation ratio involving the number of zero-valued components rather than the number of non-zeros, and is more convenient for expressing our results. A good approximation corresponds to K_d/K^* being not much larger than 1. For the cases that we analyze, we obtain upper bounds on K_d/K^* of the following form:

$$\frac{K_d}{K^*} \leq \frac{\lceil (\underline{K}+1)r \rceil - 1}{\underline{K}} \approx r, \tag{19}$$

where $\underline{K}$ is a positive integer, r is a real number greater than 1, and $\underline{K}$ and r depend on the class of instances under consideration. The approximation in (19) is justified when $\underline{K}$ is much greater than 1.

Our first result relates the quality of approximation to the condition number $\kappa(\mathbf{Q})$, defined as the ratio of the largest eigenvalue $\lambda_{\max}(\mathbf{Q})$ to the smallest eigenvalue $\lambda_{\min}(\mathbf{Q})$. Geometrically, $\kappa(\mathbf{Q})$ corresponds to the ratio between the longest and shortest axes of the ellipsoid $\mathcal{E}_\mathbf{Q}$. We expect the diagonal relaxation to be a good approximation when the condition number is low. A small value for $\kappa(\mathbf{Q})$ implies that $\mathcal{E}_\mathbf{Q}$ is nearly spherical and can therefore be enclosed by a coordinate-aligned ellipsoid $\mathcal{E}_\mathbf{D}$ of comparable size. This is illustrated in Fig. 6 in the two-dimensional case. Since $\mathcal{E}_\mathbf{Q}$ can be well-approximated by $\mathcal{E}_\mathbf{D}$ in terms of volume, one would expect a close approximation in terms of sparsity as well. We obtain an approximation guarantee in the form of (19) with $\underline{K}$ and r defined as follows:

$$\begin{aligned} \underline{K} = \underline{K}(\mathbf{S}) = \max \quad & K \quad \text{s.t.} \\ & \lambda_{\max}(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}) S_K\big(\{S_{nn}c_n^2\}\big) \leq \gamma, \\ r = r(\mathbf{S}) = \kappa(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}), & \end{aligned} \tag{20}$$

where $\mathbf{S}$ can be an arbitrary invertible diagonal matrix. With $\mathbf{S} = \mathbf{I}$, (19) states that the ratio K_d/K^* is approximately bounded by the condition number $\kappa(\mathbf{Q})$. The bound can be optimized by choosing $\mathbf{S}$ to minimize $\kappa(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1})$, i.e., as an optimal diagonal preconditioner for $\mathbf{Q}$.

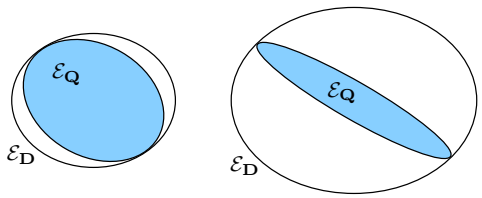

Fig. 6. Diagonal relaxations for two ellipsoids with contrasting condition numbers.

Because of space limitations, we describe only the major steps in the proof of the condition number bound above for

the case $\mathbf{S} = \mathbf{I}$. The reader is referred to [18, Sec. 3.4.4] for details. To bound the ratio K_d/K^*, we combine a lower bound $\underline{K}$ on K^* with an upper bound $\overline{K}$ on K_d. The former can be derived using the following condition [13] for the feasibility of solutions to (1) with K zero-valued components:

$$E_0(K) = \min_{|\mathcal{Z}|=K} \left\{ \mathbf{c}_{\mathcal{Z}}^T (\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}) \mathbf{c}_{\mathcal{Z}} \right\} \leq \gamma, \tag{21}$$

where $\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}$ denotes the Schur complement of $\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}$. By definition, K^* is the largest value of K for which (21) is satisfied. Hence K^* can be bounded from below by means of an upper bound on the right-hand side $E_0(K)$ in (21). Using properties of quadratic forms and Schur complements [21] and the definition of S_K we obtain

$$\begin{aligned} E_0(K) &\leq \min_{|\mathcal{Z}|=K} \left\{ \lambda_{\max}(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}) \left\| \mathbf{c}_{\mathcal{Z}} \right\|_2^2 \right\} \\ &\leq \min_{|\mathcal{Z}|=K} \left\{ \lambda_{\max}(\mathbf{Q}) \left\| \mathbf{c}_{\mathcal{Z}} \right\|_2^2 \right\} \\ &= \lambda_{\max}(\mathbf{Q}) S_K\big(\{c_n^2\}\big), \end{aligned}$$

from which it can be seen that $\underline{K}$ in (20) (with $\mathbf{S} = \mathbf{I}$) is a lower bound on K^*. Similarly, K_d is the largest K such that $E_d(K)$ in (17) is less than or equal to γ. Therefore a lower bound on $E_d(K)$ yields an upper bound on K_d. Since $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$ is a feasible solution to (17), we have $E_d(K) \geq \lambda_{\min}(\mathbf{Q}) S_K(\{c_n^2\})$ and hence $K_d \leq \overline{K} = \max\{K : \lambda_{\min}(\mathbf{Q}) S_K(\{c_n^2\}) \leq \gamma\}$. The similar expressions for $\underline{K}$ and $\overline{K}$ suggest that their ratio is approximately equal to the condition number $\kappa(\mathbf{Q})$. The detailed derivation in [18] leads to the bound in (19). The generalization to non-identity $\mathbf{S}$ is due to the invariance of (1) and (17) to arbitrary diagonal scalings. This property follows from the invariance of the zero-norm to diagonal scaling and from the ability of $\mathbf{D}$ to absorb any diagonal scalings in (17) (see [18, Sec. 3.4.3]).

Next we consider the case in which $\mathbf{Q}$ is diagonally dominant, specifically in the sense that

$$\max_m \sum_{n \neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm} Q_{nn}}} < 1, \tag{22}$$

i.e., the sum of the normalized off-diagonal entries is small in every row. In the diagonally dominant case, the diagonal relaxation is expected to provide a close approximation to the original problem. By defining $\mathcal{Z}_K$ to be the subset of indices corresponding to the K smallest values of $Q_{nn}c_n^2$, a bound of the form in (19) can be obtained with

$$\begin{aligned} &\underline{K} = \max \quad K \quad \text{s.t.} \\ &\left(1 + \max_{m \in \mathcal{Z}_K} \sum_{\substack{n \in \mathcal{Z}_K \\ n \neq m}} \frac{|Q_{mn}|}{\sqrt{Q_{mm} Q_{nn}}} \right) S_K\big(\{Q_{nn} c_n^2\}\big) \leq \gamma, \end{aligned}$$

$$r = \left(1 + \max_{m \in \mathcal{Z}_{\underline{K}+1}} \sum_{\substack{n \in \mathcal{Z}_{\underline{K}+1} \\ n \neq m}} \frac{|Q_{mn}|}{\sqrt{Q_{mm} Q_{nn}}} \right) \Bigg/ \left(1 - \max_m \sum_{n \neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm} Q_{nn}}} \right).$$

The ratio r depends on the degree of diagonal dominance of $\mathbf{Q}$ and approaches 1 as the off-diagonal entries converge to zero. The bound in (19) then implies that K_d approaches K^* as expected. The proof of the bound follows the same strategy as for the condition number bound with different expressions for $\underline{K}$ and $\overline{K}$ that reflect the diagonal dominance of $\mathbf{Q}$; for details see [18, Sec. 3.4.5].

A geometric analogue to diagonal dominance is the case in which the axes of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ are nearly aligned with the coordinate axes. Algebraically, this corresponds to the eigenvectors of $\mathbf{Q}$ being close to the standard basis vectors. More specifically, we assume that $\mathbf{Q}$ is diagonalized as $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where the eigenvalues $\lambda_n(\mathbf{Q})$ and the orthogonal matrix $\mathbf{V}$ of eigenvectors of $\mathbf{Q}$ are ordered in such a way that $\mathbf{\Delta} \equiv \mathbf{V} - \mathbf{I}$ is small. In the nearly coordinate-aligned case, we also expect a good approximation from the diagonal relaxation. If the spectral radius $\rho(\mathbf{\Delta})$ of $\mathbf{\Delta}$ is small enough to satisfy the condition $\kappa(\mathbf{Q})\rho(\mathbf{\Delta}) < 1$, then it can be shown that (19) holds with

$$\begin{aligned} &\underline{K} = \max \quad K \quad \text{s.t.} \\ &\qquad \big(1 + \kappa(\mathbf{Q})\rho(\mathbf{\Delta}) + \kappa(\mathbf{Q})\rho^2(\mathbf{\Delta})\big) S_K\big(\{\lambda_n(\mathbf{Q}) c_n^2\}\big) \leq \gamma, \\ &r = \frac{1 + \kappa(\mathbf{Q})\rho(\mathbf{\Delta}) + \kappa(\mathbf{Q})\rho^2(\mathbf{\Delta})}{1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta})}. \end{aligned}$$

The ratio r now depends on the coordinate alignment of $\mathcal{E}_{\mathbf{Q}}$ and the conditioning of $\mathbf{Q}$ and is close to 1 if $\mathcal{E}_{\mathbf{Q}}$ is nearly aligned and $\mathbf{Q}$ is well-conditioned. The proof is on similar lines as above [18, Sec. 3.4.6].

D. Numerical comparison of linear and diagonal relaxations

To complement the analysis in Sections III-B and III-C, we present in this subsection a numerical evaluation of linear and diagonal relaxations. While it was seen earlier that neither relaxation dominates the other over all possible instances of (1), the numerical comparison indicates that diagonal relaxations provide significantly stronger bounds on average in many classes of instances. The experiments also shed further light on the approximation properties of the diagonal relaxation, revealing in particular a dependence on the eigenvalue distribution of the matrix $\mathbf{Q}$.

The evaluation is conducted by generating large numbers of random instances of (1) to facilitate the investigation of properties of the relaxations. Filter design examples are considered later in Section IV. The number of dimensions N is varied between 10 and 150 and the parameter γ is normalized to 1 throughout. In the first three experiments, the eigenvectors of $\mathbf{Q}$ are chosen as an orthonormal set oriented uniformly at random over the unit sphere in N dimensions. The eigenvalues of $\mathbf{Q}$ are drawn from different power-law distributions and then

rescaled to match a specified condition number $\kappa(\mathbf{Q})$, chosen from among the values $\sqrt{N}$, N, $10N$, and $100N$. One motivation for considering power-law eigenvalue distributions stems from the typical channel frequency responses encountered in wireline communications [22]. Once $\mathbf{Q}$ is determined, each component c_n of the ellipsoid center is drawn uniformly from the interval $\left[-\sqrt{(\mathbf{Q}^{-1})_{nn}}, \sqrt{(\mathbf{Q}^{-1})_{nn}}\right]$. These bounds on c_n are in keeping with our assumption that (6) is satisfied for all n.

The linear relaxation of each instance, and more specifically the Lagrangian dual derived in [18, App. B.2], is solved using the function `fmincon` in MATLAB. We use the customized solver described in [18, Sec. 3.5] for the diagonal relaxation; a general-purpose semidefinite optimization solver such as SDPT3 [23] or SeDuMi [24] can also be used to solve (18). In addition, a feasible solution is obtained for each instance using the backward greedy algorithm of [13]. To assess the quality of each relaxation, we use the ratio of the optimal cost of the relaxation to the cost of the backward greedy solution. These ratios are denoted as R_ℓ and R_d for linear and diagonal relaxations. Since any feasible solution provides an upper bound on the optimal cost of (1), R_ℓ and R_d are lower bounds on the true approximation ratios, which are difficult to compute given the large number of instances. Note that we are returning to the original definition of approximation ratio in terms of the number of non-zero coefficients and not the number of zero-valued coefficients as in Section III-C.

In the first experiment, the eigenvalues of $\mathbf{Q}$ are drawn from a distribution proportional to $1/\lambda$, which corresponds to a uniform distribution for $\log\lambda$. While no single eigenvalue distribution can be representative of all positive definite matrices, the inverse of any positive definite matrix is also positive definite and a $1/\lambda$ eigenvalue distribution is unbiased in this regard since it is invariant under matrix inversion (up to a possible overall scaling). Fig. 7(a) plots the ratios R_ℓ and R_d as functions of N and $\kappa(\mathbf{Q})$ under a $1/\lambda$ distribution, where each point represents the average of 1000 instances. The linear relaxation approximation ratio R_ℓ does not vary much with N or $\kappa(\mathbf{Q})$. In contrast, the diagonal relaxation approximation ratio R_d is markedly higher for lower $\kappa(\mathbf{Q})$, in agreement with the association between condition number and ellipsoid sphericality and the bound in (19). Moreover, R_d also improves with increasing N so that even for $\kappa(\mathbf{Q}) = 100N$ the diagonal relaxation outperforms the linear relaxation for $N \geq 20$. The difference is substantial at large N and is reflected not only in the average ratios but also in their distributions; clear separations can be seen in [18, Sec. 3.6] between histograms of optimal values for diagonal relaxations and corresponding histograms for linear relaxations.

Figs. 7(b) and 7(c) show average approximation ratios R_ℓ and R_d for a uniform eigenvalue distribution and a $1/\lambda^2$ distribution respectively. It is straightforward to show that a $1/\lambda^2$ distribution for the eigenvalues of $\mathbf{Q}$ corresponds to a uniform distribution for the eigenvalues of $\mathbf{Q}^{-1}$. The behavior of R_ℓ is largely unchanged. Each R_d curve in Fig. 7(b) however is lower than its counterpart in Fig. 7(a) and the dependence of R_d on the condition number is more pronounced. The linear

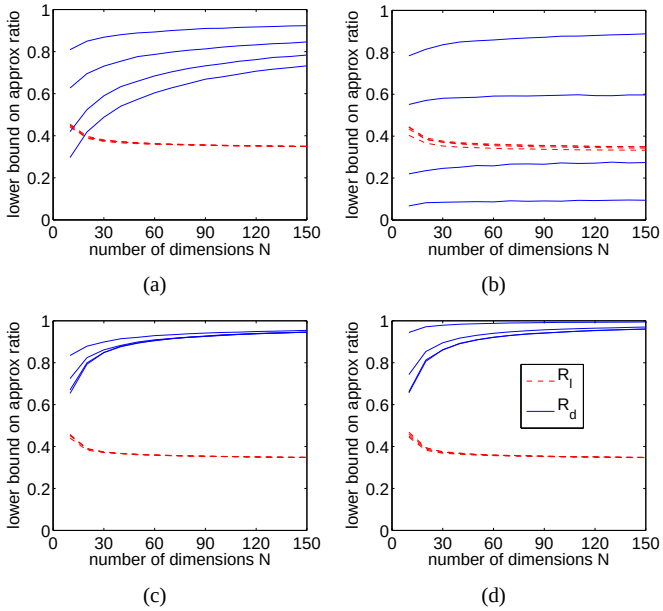

Fig. 7. Average approximation ratios R_ℓ and R_d for (a) a $1/\lambda$ eigenvalue distribution, (b) a uniform eigenvalue distribution, (c) a $1/\lambda^2$ eigenvalue distribution, and (d) exponentially decaying $\mathbf{Q}$ matrices. In (a)–(c), $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom within each set of curves. In (d), $\rho = 0.1, 0.5, 0.9, 0.99$ from top to bottom within each set of curves.

relaxation is now preferable to the diagonal relaxation when $\kappa(\mathbf{Q})$ is significantly greater than N. On the other hand, the R_d curves in Fig. 7(c) are higher than in Figs. 7(a) and 7(b) and the dependence on $\kappa(\mathbf{Q})$ is reduced.

The differences among Figs. 7(a)–(c) suggest that the diagonal relaxation yields a better approximation when the eigenvalue distribution of $\mathbf{Q}$ is weighted toward lower values, as in Figs. 7(a) and 7(c), so that most of the eigenvalues are small and of comparable size. While a rigorous explanation for this dependence on eigenvalue distribution is a subject for future study, the dependence can be explained more informally by utilizing the inverse relationship between eigenvalues and axis lengths of the ellipsoid $\mathcal{E}_\mathbf{Q}$, combined with the following geometric intuition: Assuming that $\mathcal{E}_\mathbf{Q}$ is not close to spherical, i.e., $\kappa(\mathbf{Q})$ is relatively large, it is preferable for most of the ellipsoid axes to be long rather than short, and for the long axes to be comparable in length. Such an ellipsoid tends to require a comparatively smaller coordinate-aligned enclosing ellipsoid, and consequently the diagonal relaxation tends to be a better approximation. For example, in three dimensions, a severely oblate spheroid can be enclosed in a smaller coordinate-aligned ellipsoid on average than an equally severely prolate spheroid.

In a fourth experiment, $\mathbf{Q}$ is chosen to correspond to an exponentially decaying autocorrelation function with entries given by $Q_{mn} = \rho^{|m-n|}$, where the decay ratio ρ is varied between 0.05 and 0.99. The vector $\mathbf{c}$ is generated as before based on the diagonal entries of $\mathbf{Q}^{-1}$. It can be shown that only positive values of ρ need to be considered since ρ and $-\rho$ are equivalent in terms of the zero-norm cost [18, Sec. 3.6]. In addition, $\mathbf{Q}$ is diagonally dominant in the sense of (22) for $\rho \leq 1/3$. Fig. 7(d) shows the approximation ratios R_ℓ

and R_d for four values of ρ, averaged over 1000 instances as before. As with the condition number $\kappa(\mathbf{Q})$ in Figs. 7(a)–(c), the decay ratio ρ does not appear to have much effect on R_ℓ. Furthermore, while the analysis in Section III-C predicts a close approximation from the diagonal relaxation for $\rho = 0.1$, it is somewhat surprising that the performance does not degrade by much even for ρ close to 1.

The results in Fig. 7 indicate that diagonal relaxations result in better bounds than linear relaxations in many instances. This can be true even when the condition number $\kappa(\mathbf{Q})$ or the decay ratio ρ is high, whereas the analysis in Section III-C is more pessimistic. The experiments also confirm the dependence of the diagonal relaxation on the conditioning and diagonal dominance of $\mathbf{Q}$ and indicate an additional dependence on the eigenvalue distribution.

IV. Design examples

In this section, the design examples in [13] are used to evaluate the complexity of variants of our branch-and-bound algorithm employing either linear relaxations, diagonal relaxations, or no relaxations. We also compare our algorithm to the commercial mixed-integer programming solver CPLEX 12.4 [16]. The results demonstrate that the proposed techniques, in particular diagonal relaxations, can significantly reduce the complexity of an exact solution to (1), especially in more difficult instances where order-of-magnitude decreases are possible. For instances that are too difficult to be solved exactly in the allotted time, diagonal relaxations yield tighter bounds on the deviation from optimality of the final solution.

Our branch-and-bound algorithm is implemented in MATLAB 7.11 (2010b), including solvers for linear and diagonal relaxations as described in Section III-D. For CPLEX, we use the mixed integer formulation of the problem (more precisely the split-variable formulation leading to (12)), which is passed to the CPLEX MEX executable with default solver options. The experiments are run on a 2.4 GHz quad-core Linux computer with 3.9 GB of memory. Our algorithm uses more than one core only when the dimension of the computation exceeds 50 or so; CPLEX however is able to exploit all four cores all the time. Complexity is measured in terms of running time and the number of subproblems processed.

A. Wireless channel equalization

The first example involves the design of sparse equalizers for a high-definition television terrestrial broadcast channel. The multipath parameters for the channel are given in Table I, where the delays τ_i are expressed as multiples of the sampling period. Following [6], [7], the transmit and receive filters are chosen to be square-root raised-cosine filters with excess bandwidth parameter $\beta = 0.115$. The transmitted sequence and noise are assumed to be white with the ratio of the signal variance to the noise variance, i.e., the input SNR, set to 10 dB. The number of non-zero equalizer coefficients is minimized subject to a constraint on the allowable MSE δ. The formulation of the sparse equalizer design problem in the form of (1) is discussed in [13]. The solution time is limited to 10^5 seconds for all algorithms.

TABLE I
Multipath parameters for the HDTV equalization example.

i	0	1	2	3	4	5
τ_i	0	4.84	5.25	9.68	20.18	53.26
a_i	0.5012	-1	0.1	0.1259	-0.1995	-0.3162

Table II shows the final cost values, solution times, and numbers of subproblems for equalizer lengths $N = L + 1, 1.5L + 1, 2L + 1$, where $L = 54$ is the largest channel delay, and MSE values up to 2 dB above the minimum MSE $\delta_{\min}$ corresponding to the optimal non-sparse equalizer. Note that for $N = 2L + 1$, an exhaustive search would require considering $2^{109} \approx 6 \times 10^{32}$ configurations. The final cost shown in Table II is the optimal cost when at least one of the algorithm variants converges within the allowed solution time; otherwise it is the cost of the incumbent solution at termination. For instances in which the algorithm does not converge, the final optimality gap, i.e., the difference between the incumbent cost and the smallest of the lower bounds for open subproblems, is shown in parentheses in place of the solution time.

We focus first on the three variants of the proposed algorithm, which are all implemented in MATLAB and thus directly comparable in terms of solution time. Table II shows that the use of diagonal relaxations can dramatically reduce complexity relative to the other two variants, particularly for the more difficult instances at larger lengths and intermediate MSE. Intermediate MSE values pose a greater difficulty because the sparsity level also tends to be intermediate and the number of configurations with the same sparsity, i.e., the binomial coefficient $\binom{N}{K}$, is very large. Diagonal relaxations become instrumental for improving lower bounds and pruning large numbers of subproblems. For instances that cannot be solved exactly within the given time, diagonal relaxations result in smaller optimality gaps.

In contrast, for the easier instances at shorter lengths and higher MSE, the algorithm variant that avoids relaxations is the most efficient. In these cases, either the dimension or the incumbent cost is low enough for the infeasibility bounds in Section III-A to be effective, and consequently the added effort of solving relaxations is not justified. The $N = 55$, $\delta/\delta_{\min} = 0.02$ dB instance can also be solved efficiently without relaxations because a significant fraction of the coefficients cannot take zero values and are thus eliminated as discussed in Section III-A. For $\delta/\delta_{\min} = 0.02$ dB and $N = 82, 109$ however, diagonal relaxations still yield substantial savings.

Linear relaxations are not observed to reduce solution times except in the most difficult instances where the modest improvement in lower bounds is still valuable. The linear relaxation variant is faster than the diagonal relaxation variant only at high MSE where both relaxations are unnecessary but the overhead of solving linear relaxations is smaller.

As for the number of subproblems, Table II shows that when all three variants of our algorithm converge, the one using diagonal relaxations solves substantially fewer subproblems. However, when some or all of the variants fail to converge,

TABLE II
COMPLEXITY OF DIFFERENT BRANCH-AND-BOUND ALGORITHMS FOR THE EQUALIZATION EXAMPLE. NUMBERS IN PARENTHESES INDICATE THE FINAL OPTIMALITY GAP IN CASES OF NON-CONVERGENCE.

N	$\delta/\delta_{\min}$ [dB]	final cost	time [s] (gap)				number of subproblems			
			none	linear	diagonal	CPLEX	none	linear	diagonal	CPLEX
55	0.02	43	0.43	1.40	0.56	19.50	750	750	734	7370
	0.05	36	17.3	31.1	6.6	191.5	9492	9236	3890	72256
	0.1	28	7.7	35.2	5.0	153.2	6688	3588	712	42466
	0.2	20	0.65	7.91	1.19	50.85	1492	698	88	8503
	0.4	13	0.15	2.88	1.30	16.80	406	302	74	2463
	0.7	8	0.08	1.32	0.81	9.93	166	144	40	1149
	1.0	5	0.014	0.141	0.166	1.522	14	12	4	5672
	1.5	3	0.011	0.028	0.040	0.425	0	0	0	199
	2.0	2	0.002	0.002	0.002	0.182	0	0	0	22
82	0.02	63	75	101	8.5	473	16134	15770	3238	113501
	0.05	55	80793	27568	801	23081	506836	279986	37234	3837752
	0.1	47	(5)	(3)	97621	(2)	339516	290058	543652	10863093
	0.2	34	(2)	15217	1057	39982	338074	203282	39622	4446718
	0.4	22	330	137	63	1126	35414	9000	1942	121759
	0.7	14	7.6	28.6	21.1	206.0	4996	2098	454	17813
	1.0	10	0.9	8.9	10.4	80.0	1410	642	196	7887
	1.5	5	0.041	0.346	0.758	106.26	34	22	14	130522
	2.0	3	0.024	0.043	0.075	0.779	0	0	0	567
109	0.02	85	4242	2576	39	3838	104946	75962	4894	624995
	0.05	76	(5)	(3)	10131	(2)	317473	270808	148652	7572442
	0.1	67	(9)	(7)	(3)	(7)	298086	245448	223834	6902299
	0.2	56	(14)	(10)	(6)	(12)	280963	234673	210940	5620187
	0.4	38	(9)	(6)	(3)	(6)	288158	214381	217697	5242439
	0.7	25	(5)	(2)	37185	(2)	212572	262994	243496	5453732
	1.0	17	45428	2466	925	14783	347892	61334	12360	889357
	1.5	10	22.0	40.7	67.4	795.9	7632	2420	774	40677
	2.0	5	0.09	0.66	2.56	20.78	60	34	24	569

the optimality gap becomes the more important statistic since the number of subproblems may simply reflect the amount of computation performed in the allotted time. For $N = 82$ and $\delta/\delta_{\min} = 0.1$ dB, the diagonal relaxation variant actually solves more subproblems but converges in the end. This may be interpreted as evidence that the algorithm has moved beyond the higher-dimensional subproblems that slowed progress for the other two variants.

In comparison to CPLEX, Table II shows that the diagonal relaxation variant of our algorithm is much more efficient. Indeed, the other two variants are also more efficient than CPLEX in easier instances, whereas in difficult instances CPLEX becomes comparable to the linear relaxation variant. These favorable comparisons are obtained despite CPLEX's advantages as a compiled executable capable of full multicore execution. The computational advantage of CPLEX can be seen in the number of subproblems processed per unit time, which in more difficult instances is generally an order of magnitude higher than for our algorithm. It is difficult to identify precisely the reasons for the relative inefficiency of CPLEX given its use of many additional techniques beyond basic branch-and-bound. We have observed that the heuristic used by CPLEX is less effective than the backward selection heuristic used in our algorithm. To obtain lower bounds, CPLEX uses linear relaxations and may solve too many of them in the sense of not improving bounds, in contrast to our more judicious approach (see the conditions on Step 5 in Fig. 3). CPLEX is likely not able to use diagonal relaxations or the infeasibility bounds in Section III-A, which are specific to our problem. The infeasibility bounds in particular can eliminate many infeasible subproblems and improve bounds incrementally with minimal computation, and can also reduce subproblem dimensions as discussed in Section II.

The benefits of solving diagonal relaxations in this example can be partly attributed to the properties of the matrix $\mathbf{Q}$, which is largely determined by the channel response. In a multipath environment with a sparse set of arrivals, the resulting matrix $\mathbf{Q}$ tends to be well-conditioned with the largest entries near the diagonal, although the strict definition of diagonal dominance in (22) is not satisfied in this example.

B. MVDR beamforming

In a second example, we turn to the design of sparse minimum-variance distortionless-response (MVDR) beamformers for signal detection. Since the current branch-and-bound algorithm is intended for real-valued filter design, we focus on a real-valued formulation of the MVDR beamforming problem. The complex-valued generalization of the branch-and-bound algorithm is a subject for future study. We consider an environment with two discrete interference sources at angles θ_1 and θ_2 from the array axis, where $\cos\theta_1 = 0.18$ and $\cos\theta_2 = 0.73$, together with isotropic (white) noise. The target direction θ_0 is swept over 140 values from $\cos\theta_0 = 0$ to $\cos\theta_0 = 1$. The interferer powers are set at 10 and 25 dB respectively relative to the white noise power, while the signal power is normalized to unity. The number of non-zero array weights is fixed at 30 and four array lengths $N = 30, 40, 50, 60$ are considered. Further details of the experiment can be found in [13].

For each length N and target angle θ_0, the objective is to maximize the output SNR, defined as the ratio between the mean array output and the standard deviation. For $N = 30$, the SNR is maximized by the conventional non-sparse MVDR beamformer. For $N = 40, 50, 60$, a linear search over SNR is performed, starting from the highest SNR achieved at the next lowest value of N and increasing in 0.05 dB increments. At a fixed SNR, the minimization of the number of non-zero weights corresponds to an instance of problem (1), as shown in [13]. For this example, it suffices to test for feasibility at each SNR, i.e., to determine whether a solution with 30 non-zero weights exists subject to the SNR constraint. We use branch-and-bound for this purpose, terminating it as soon as such a solution is found, or alternatively as soon as all of the subproblem lower bounds rise above 30, indicating infeasibility. We compare the three variants of our algorithm as in Section IV-A, allowing one hour of processing time per SNR value, but do not include CPLEX because of its relative inefficiency. In cases where neither of the terminating conditions is met within one hour, we obtain bounds on the maximum achievable SNR at the current (N, θ_0) pair instead of a definite value. The lower bound corresponds to the highest SNR at which the algorithm is able to find a feasible solution, while the upper bound corresponds to the lowest SNR at which the algorithm is able to prove infeasibility.

Table III summarizes the results of the algorithm comparison. Instances are divided into two groups depending on whether the algorithm variant converged within the allowed time, and average statistics within each group are reported. For $N = 40$, the vast majority of instances are simple enough to be solved most efficiently without relaxations. For $N = 50$ and 60 however, diagonal relaxations reduce the average solution time and number of subproblems by an order of magnitude in instances in which the algorithm converges. Indeed for $N = 60$, the algorithm fails to converge in a large majority of instances unless diagonal relaxations are used, in which case the opposite is true. Diagonal relaxations also yield tighter bounds on the optimal SNR in cases of non-convergence as measured by the gap between the upper and lower bounds. Linear relaxations on the other hand offer no benefits in this example. We note that the instances in which the diagonal relaxation variant does not converge tend to correspond to target array manifold vectors with nearly equal components, leading to a large number of array configurations with similar SNR and thus complicating the branch-and-bound search.

As in the channel equalization example, the properties of the present beamforming example favor the use of diagonal relaxations. Specifically, the matrix $\mathbf{Q}$ has two large eigenvalues corresponding to the interferers while the remaining eigenvalues are small and equal, corresponding to white noise. As discussed in Section III-D, diagonal relaxations tend to result in good approximations for this type of eigenvalue distribution even though the condition number may be high.

V. Conclusions and future work

We have proposed a branch-and-bound algorithm for designing maximally sparse filters subject to a quadratic constraint on filter performance, with particular emphasis on the determination of lower bounds on the optimal cost. Low-complexity bounds based on infeasibility can be easily incorporated into branch-and-bound to yield incremental improvement, while stronger bounds can be obtained through linear and diagonal relaxations, both of which involve convex optimization and are therefore efficient. Filter design examples demonstrated that substantial complexity reductions can be achieved with diagonal relaxations in particular, especially in more difficult instances and even when comparing a MATLAB implementation to commercial software. In the case of early termination, solving diagonal relaxations leads to tighter bounds on the deviation from optimality. The techniques in this paper make optimal design more accessible not only to filter designers but also developers of design algorithms. Specifically, the proposed branch-and-bound algorithm can be used to more easily evaluate lower-complexity approximate algorithms, as we have done for the backward greedy algorithm in [13].

Our positive experience with diagonal relaxations inspires interest in the general approach of exploiting an efficiently solvable special case of a problem to approximate a broader class. A potential next candidate is the tridiagonal case discussed in [13]. A similar approach has been applied to design filters with efficient binary representations [18] and could be extended to sparse filter design under a minimax constraint on the frequency response [4]. Future work could also be directed at more sophisticated implementations of the branch-and-bound algorithm. Our experience with the current MATLAB implementation suggests that for filters of length up to 100, optimality can be certified within a few hours on a present-day computer, a figure that would likely be improved by implementing the algorithm or critical parts of it in a more efficient programming language such as C. Branch-and-bound is also highly parallelizable and could thus benefit from multiprocessor and cloud computing.

References

[1] J. T. Kim, W. J. Oh, and Y. H. Lee, "Design of nonuniformly spaced linear-phase FIR filters using mixed integer linear programming," *IEEE Trans. Signal Process.*, vol. 44, pp. 123–126, Jan. 1996.

[2] Y.-S. Song and Y. H. Lee, "Design of sparse FIR filters based on branch-and-bound algorithm," in *Proc. Midwest Symp. Circuits. Syst.*, vol. 2, Aug. 1997, pp. 1445–1448.

[3] D. Mattera, F. Palmieri, and S. Haykin, "Efficient sparse FIR filter design," in *Proc. ICASSP*, vol. 2, May 2002, pp. 1537–1540.

[4] T. Baran, D. Wei, and A. V. Oppenheim, "Linear programming algorithms for sparse filter design," *IEEE Trans. Signal Process.*, vol. 58, pp. 1605–1617, Mar. 2010.

[5] S. A. Raghavan, J. K. Wolf, L. B. Milstein, and L. C. Barbosa, "Non-uniformly spaced tapped-delay-line equalizers," *IEEE Trans. Commun.*, vol. 41, no. 9, pp. 1290–1295, Sep. 1993.

[6] I. J. Fevrier, S. B. Gelfand, and M. P. Fitz, "Reduced complexity decision feedback equalization for multipath channels with large delay spreads," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 927–937, Jun. 1999.

[7] F. K. H. Lee and P. J. McLane, "Design of nonuniformly spaced tapped-delay-line equalizers for sparse multipath channels," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 530–535, Apr. 2004.

[8] H. Sui, E. Masry, and B. D. Rao, "Chip-level DS-CDMA downlink interference suppression with optimized finger placement," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3908–3921, Oct. 2006.

[9] G. Kutz and D. Raphaeli, "Determination of tap positions for sparse equalizers," *IEEE Trans. Commun.*, vol. 55, no. 9, pp. 1712–1724, Sep. 2007.

TABLE III
EFFECT OF RELAXATIONS ON BRANCH-AND-BOUND COMPLEXITY FOR THE BEAMFORMING EXAMPLE.

		converged instances			non-converged instances		
N	relaxation	# instances	avg time [s]	avg # subproblems	# instances	avg gap [dB]	avg # subproblems
40	none	137	0.20	167	3	0.53	12.6×10^5
	linear	137	1.13	167	3	0.53	11.3×10^5
	diagonal	137	1.22	93	3	0.23	3.7×10^5
50	none	124	158	20.3×10^3	16	0.42	7.6×10^5
	linear	124	195	19.4×10^3	16	0.41	6.3×10^5
	diagonal	133	27	2.7×10^3	7	0.14	2.6×10^5
60	none	15	417	28.7×10^3	125	0.40	6.7×10^5
	linear	18	695	23.8×10^3	122	0.39	4.9×10^5
	diagonal	131	15	1.4×10^3	9	0.28	3.3×10^5

[10] A. Gomaa and N. Al-Dhahir, "A new design framework for sparse FIR MIMO equalizers," *IEEE Trans. Commun.*, 2011, to appear.

[11] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen, "Sparse linear prediction and its applications to speech processing," *IEEE Audio, Speech, Language Process.*, vol. 20, no. 5, pp. 1644–1657, Jul. 2012.

[12] C. K. Sestok, "Data selection for detection of known signals: The restricted-length matched filter," in *Proc. ICASSP*, vol. 2, May 2004, pp. 1085–1088.

[13] D. Wei, C. K. Sestok, and A. V. Oppenheim, "Sparse filter design under a quadratic constraint: Low-complexity algorithms," submitted to IEEE Transactions on Signal Processing, 2012.

[14] D. Bertsimas and R. Weismantel, *Optimization Over Integers*. Belmont, MA: Dynamic Ideas, 2005.

[15] M. H. Hayes, *Statistical digital signal processing and modeling*. New York: John Wiley & Sons, 1996.

[16] *IBM ILOG CPLEX 12.4 User's Manual*, IBM ILOG, 2012.

[17] D. Wei and A. V. Oppenheim, "Sparsity maximization under a quadratic constraint with applications in filter design," in *Proc. ICASSP*, Mar. 2010, pp. 117–120.

[18] D. Wei, "Design of discrete-time filters for efficient implementation," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, May 2011.

[19] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Nashua, NH: Athena Scientific, 1997.

[20] C. Lemaréchal and F. Oustry, "Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization," INRIA, Tech. Rep. RR-3710, 1999.

[21] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, UK: Cambridge University Press, 1994.

[22] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding digital subscriber line technology*. Prentice Hall, 1999.

[23] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 — a MATLAB software package for semidefinite programming," *Optim. Method. Softw.*, vol. 11, pp. 545–581, 1999, latest version available at http://www.math.nus.edu.sg/ mattohkc/sdpt3.html.

[24] J. F. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," *Optim. Method. Softw.*, vol. 11, pp. 625–653, 1999.

Dennis Wei (S'09–M'11) received S.B. degrees in electrical engineering and in physics in 2006, the M.Eng. degree in electrical engineering in 2007, and the Ph.D. degree in electrical engineering in 2011, all from the Massachusetts Institute of Technology. He is currently a post-doctoral research fellow in the Department of Electrical Engineering and Computer Science at the University of Michigan. His research interests lie broadly in signal processing, optimization, and statistical inference and learning. Areas of focus include adaptive sensing and processing, filter design, and non-uniform sampling.

Dr. Wei is a member of Phi Beta Kappa, Sigma Xi, Eta Kappa Nu, and Sigma Pi Sigma. He has been a recipient of the William Asbjornsen Albert Memorial Fellowship at MIT and a Siebel Scholar.

Alan V. Oppenheim (M'65–SM'71–F'77–LF'03) was born in New York, New York on November 11, 1937. He received S.B. and S.M. degrees in 1961 and an Sc.D. degree in 1964, all in Electrical Engineering, from the Massachusetts Institute of Technology. He is also the recipient of an honorary doctorate from Tel Aviv University.

In 1964, Dr. Oppenheim joined the faculty at MIT, where he is currently Ford Professor of Engineering. Since 1967 he has been affiliated with MIT Lincoln Laboratory and since 1977 with the Woods Hole Oceanographic Institution. His research interests are in the general area of signal processing and its applications. He is coauthor of the widely used textbooks Discrete-Time Signal Processing (now in its third edition), Signals and Systems and Digital Signal Processing. He is also editor of several advanced books on signal processing and coauthor of the text Signals, Systems, and Inference, published online through MITs OpenCourseWare.

Dr. Oppenheim is a member of the National Academy of Engineering, a fellow of the IEEE, and a member of Sigma Xi and Eta Kappa Nu. He has been a Guggenheim Fellow and a Sackler Fellow. He has received a number of awards for outstanding research and teaching, including the IEEE Education Medal, the IEEE Jack S. Kilby Signal Processing Medal, the IEEE Centennial Award and the IEEE Third Millennium Medal. From the IEEE Signal Processing Society he has been honored with the Education Award, the Society Award, the Technical Achievement Award and the Senior Award. He has also received a number of awards at MIT for excellence in teaching, including the Bose Award, the Everett Moore Baker Award, and several awards for outstanding advising and mentoring.